

A DISTRIBUTED FLIGHT SOFTWARE DESIGN FOR SATELLITE FORMATION FLYING CONTROL

Joseph B. Mueller and Margarita Brito
Princeton Satellite Systems
33 Witherspoon Street
Princeton, NJ 08542
{jmueller, megui}@psatellite.com

ABSTRACT

Several NASA and DoD missions are envisioned that will utilize distributed, autonomous clusters of spacecraft. The Air Force Research Laboratory initiated the TechSat 21 mission to demonstrate the key enabling technologies of formation flying and distributed radar. Princeton Satellite Systems developed the Formation Flying Module (FFM) for TechSat 21 to provide autonomous reconfiguration, formation keeping, and collision avoidance capabilities to the three-satellite cluster. The process of developing flight software for such a distributed system has brought to light significant design challenges. Examples include developing a cluster-level fault management plan, designing an autonomous control system which respects the various constraints imposed by the spacecraft design, and defining a sensible ground command interface to the cluster. These challenges are likely to remain important issues for future missions, especially as the complexity and size of the cluster grows. This paper presents an overview of the FFM design along with the motivations and challenges associated with the design process.

1. INTRODUCTION

There is a growing interest in many organizations, including NASA and the Department of Defense, to use cooperative fleets of autonomous spacecraft to accomplish complex mission objectives. The motivations for the distributed satellite paradigm are often linked to the physical requirements of the mission, such as synthetic aperture radar. Additional benefits include greater redundancy, improved performance and reduced cost.

The attractive qualities of a formation flying solution come at the cost of significant design challenges which should not be overlooked. Primary drawbacks with the distributed satellite approach include significant dependence upon communication, the risk of collision, and increased demands on fuel to maintain tight relative position control. Furthermore, the proper coordination and management of a satellite cluster is a challenge in and of itself. The complexity

of the overall mission is increased substantially as the number of satellites in the cluster grows. It is certain that the potential benefits can be fully achieved only if the onboard software is sufficiently robust and autonomous.

Princeton Satellite Systems has been working for several years to address the autonomy-related challenges of formation flying. A key component of this research has been the development of ObjectAgent™. The ObjectAgent system has been developed under Air Force Research Laboratory (AFRL) Phase II Small Business Innovative Research (SBIR) funding. Development continues under the Cross-Enterprise Technology Development Program and through internal IR&D funding. A brief description of ObjectAgent is provided in Section 3.

Previous papers have addressed the MATLAB prototyping and real-time C++ development of ObjectAgent, and have described the research into agent organizations for distributed satellite control^{1,2,3,4,5,6}. These papers have also described the various multi-agent, multi-satellite simulations that have been assembled.

This paper describes a specific ObjectAgent-based design of a distributed, formation flying control system, and highlights both the engineering and software-level design issues that have surfaced. The following section describes the formation flying operations that were envisioned for TechSat 21. Section 3 provides an overview of the Formation Flying Module as implemented in ObjectAgent. Section 4 describes the design of the formation flying control system, while Section 5 details the fault management design. Finally, the current status of the Formation Flying Module is presented and directions for future work are provided.

2. FORMATION FLYING OPERATIONS

The TechSat 21 cluster was planned to consist of three identical spacecraft, each with a mass of 180 kg. They were to be launched on the same rocket and inserted into a circular orbit at an altitude of 550 km and a 34.5 degree inclination. The spacecraft design

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2003		2. REPORT TYPE		3. DATES COVERED 00-00-2003 to 00-00-2003	
4. TITLE AND SUBTITLE A Distributed Flight Software Design for Satellite Formation Flying Control				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Princeton Satellite Systems, 33 Witherspoon Street, Princeton, NJ, 08542				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT see report					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 11	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

is 3-axis stabilized, using a star tracker, sun sensor, and magnetometer for attitude determination and reaction wheels and torque rods for attitude control. Each spacecraft was to be equipped with an inter-satellite link (ISL), which provides communication as well as range / range-rate measurements. The relative navigation unit was designed to use the ISL along with an onboard propagator and a GPS receiver to estimate the relative position and velocity. The relative motion of each satellite was to be controlled by means of a single Hall-effect thruster (HET), which nominally produces 11.4 mN of thrust.

The main objectives of the mission were to perform distributed radar experiments and to conduct formation flying operations. In many cases, sustained formation flying control would be required before the radar experiment could be conducted.

The scope of the planned formation flying operations is best described by the mission profile. According to the nominal profile, the cluster was to be initialized to a leader-follower formation, with approximately 5 km of separation between the spacecraft. Over several weeks, the separation distance was to be gradually reduced to 100 m. At this point, the cluster would reconfigure to achieve relative elliptical motion, or a "football orbit". The size of the relative ellipse would then be gradually increased over several weeks. Finally, an out-of-plane component could be added, depending upon fuel availability.

Before designing the control system, it is first important to understand the manner in which the mission developers wish to command the cluster. The command interface to the cluster represents a significant portion of the control and software design effort, and should be fully examined prior to developing a control strategy. In general, three types of formation control are desired:

- 1) ground-commanded reconfigurations
- 2) autonomous formation keeping
- 3) autonomous collision avoidance

In a reconfiguration, the ground-station identifies the desired type of relative motion for each spacecraft, and the control system computes the delta-v's to achieve that motion. Autonomous formation keeping involves intermittently applying orbit corrections in order to reject disturbances and maintain the desired relative motion. Collision avoidance maneuvers are planned only in response to cluster-level failures that result in a significant probability of collision.

The first step in defining the ground-cluster interface is defining the relative frame. At all times, one of the spacecraft in the cluster is specified as the reference, which defines the origin of the relative frame. The

remaining two spacecraft are considered *relatives*. The designation of the reference spacecraft may be changed at any time by the ground.

A particular formation is defined by expressing the desired relative motion of the *relative* satellites. A special coordinate system was developed in which the ground-station can express this desired relative motion. The chosen coordinate system consists of a set of static geometric goals. The goal set consists of the following five parameters:

- y_0 Along-track offset
- a_E Semi-major axis of the relative ellipse
- \square_0 Phase angle on relative ellipse at the ascending equator crossing
- z_i Cross-track amplitude due to an inclination difference
- z_\square Cross-track amplitude due to a right ascension difference

This static geometric goal set defines the desired relative state of a spacecraft with respect to the reference throughout the entire course of its orbit. It is worth noting that the goals have only 5 degrees of freedom as opposed to 6. This is because a constraint of zero secular drift is imposed. The diagram in Figure 1 illustrates the relative motion described by the above parameters in Hill's frame. In this frame, \mathbf{x}_H points in the radial direction, \mathbf{y}_H along the velocity vector, and \mathbf{z}_H in the orbit-normal direction.

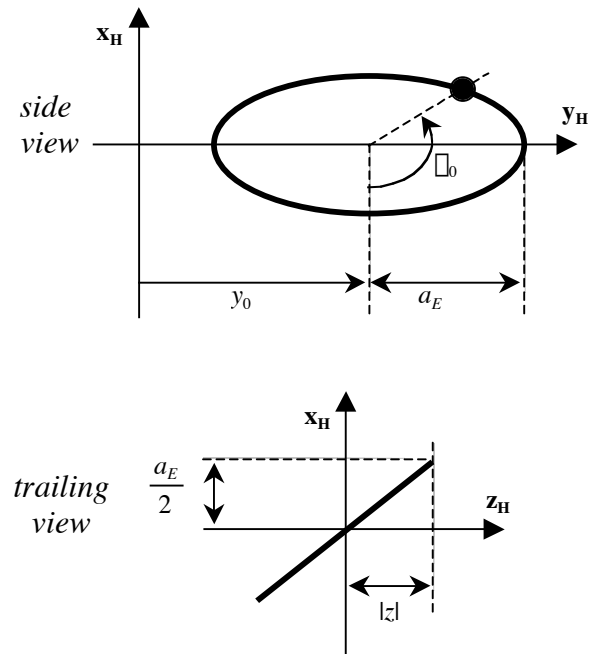


Figure 1: Relative Motion of Geometric Goals

Here, the distance labeled $|z|$ is the total cross-track amplitude,

$$|z| = \sqrt{z_i^2 + z_0^2} \quad (1)$$

In general, the motion can have three distinct attributes: 1) along-track offset, 2) relative elliptical motion, and 3) out-of-plane component. These three attributes can be combined in a total of six different ways. This has led to the formal definition of the following 6 types of relative motion.

Table 1: Types of Relative Motion

Type	Description	Restriction
IPLF	In-Plane Leader-Follower	$ z = 0$ $a_E = 0$
CIPE	Centered In-Plane Ellipse	$ z = 0$ $y_0 = 0$
NCIPE	Non-centered In-Plane Ellipse	$ z = 0$
OOPLF	Out-of-plane Leader-Follower	$a_E = 0$
COOPE	Centered Out-of-plane Ellipse	$y_0 = 0$
NCOOPE	Non-Centered Out-of-plane Ellipse	<i>none</i>

It should be noted that this geometric goal set may also be applied to eccentric orbits. The goals provided by the ground would correspond to a particular point in the orbit, and the desired parameter values would vary with the argument of latitude.

3. SOFTWARE ARCHITECTURE

One of the objectives of TechSat 21 was to provide an autonomous on-board Cluster Manager which would enable a cluster of satellites to function as a single *virtual* satellite. The Cluster Manager would handle all cluster level commanding and telemetry, implementing formation flying control and cluster level fault management, providing knowledge sharing and health summarization for the cluster, and providing payload control.

The Cluster Manager would reside on each spacecraft along with traditional flight software, as shown in Figure 2. The Spacecraft Command Language (SCL) provides the infrastructure, performing command and data handling, command execution, and managing a shared database of telemetry across the cluster. The Software Bus provides a connection between SCL and the other software modules of the Cluster Manager.

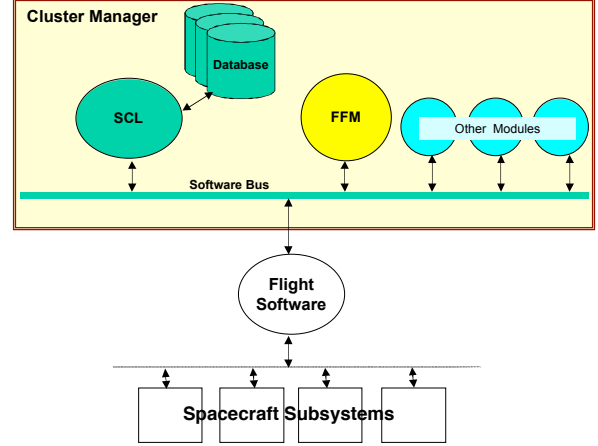


Figure 2: Cluster Manager Organization

The Formation Flying Module (FFM) is the component of the Cluster Manager that provides formation flying control capability to the cluster. The FFM has been designed and implemented within the ObjectAgent environment. It is appropriate to first provide an overview of ObjectAgent before proceeding to the discussion of the FFM architecture.

ObjectAgent is an agent-based software architecture for the control of autonomous, distributed systems. Agents are the main "software units," used to organize and implement all of the functionality. Each agent is implemented as a separate thread, has a fully-defined set of inputs and outputs, and is responsible for carrying out a specific task. Communication between agents is provided through the PostOffice – a powerful messaging system modeled after the internet. The architecture includes built-in support for TCP/IP and has an extensible framework for supporting other protocols.

Formation Flying Module

The FFM design consists of 10 agents, which may be classified into three different categories as shown in Figure 3. The module is connected to other software components on the spacecraft, including the attitude determination and control system (ADCS), the relative navigation system, the HET and the ISL. This connection is made through the Software Bus.

The architecture is distributed in that a separate instance of the FFM exists and runs onboard each spacecraft in the cluster. Communication between modules on separate spacecraft is achieved via the ISL.

A centralized management and control approach is used. Maneuver planning for the cluster is performed onboard a single spacecraft, designated as the

supervisor. The other satellites are considered *subordinates*. The designation of the supervisor spacecraft may be changed at any time by the ground.

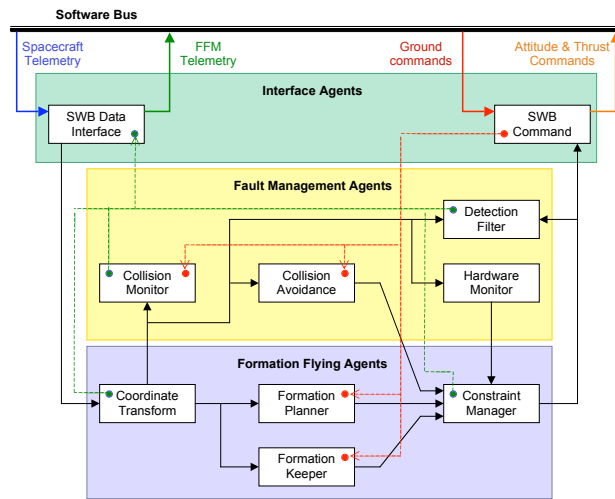


Figure 3: Agent Architecture of the FFM

Interface Agents

The interface agents are responsible for handling the interface between ObjectAgent and the Software Bus. Their primary role is to convert between Software Bus messages and ObjectAgent messages.

The *SWB Command Interface* agent has two roles. One is to receive all formation flying commands supplied from the ground, and route them to the appropriate agent(s). The other is to receive hardware commands from the Formation agent, and execute the commands using predefined scripts.

The primary role of the *SWB Data Interface* is to retrieve a variety of telemetry data from other subsystems, and route the information to the appropriate agent(s). It also receives periodic telemetry messages from the FFM agents, and supplies the data to the software bus for future downlink.

Fault Management Agents

The fault management agents implement three different aspects of fault management for the cluster.

- Collision Monitoring and Avoidance
- Thruster Fault Detection
- Maneuver Constraint Identification

The collision monitoring and avoidance is carried out by the *Collision Monitor* and *Collision Avoidance*

agents. The *Collision Monitor* uses guaranteed set membership methods to predict spacecraft's positions forward in time⁹. This agent updates every second and computes the probability of intersection with one of the other spacecraft in the cluster. If the probability of collision exceeds the designated threshold (currently set at 1%), it notifies the *Collision Avoidance* agent with the expected time of collision.

The *Collision Avoidance* agent remains inactive until notified of a potential collision by the *Collision Monitor* agent. At this point, it plans an avoidance maneuver. The avoidance planning approach was developed according to the following objectives:

1. Move one satellite away from the collision point
2. Expend a limited amount of delta-v
3. Do not drift away from the cluster

Based on these objectives, a simple two-burn maneuver was chosen, consisting of equal and opposite burns applied in the along-track direction. Only the satellite which has detected the potential collision performs a maneuver. The length of each burn is subject to a maximum delta-v parameter which may be changed on-orbit at any time. The time of the burns is chosen such that a maximum amount of radial offset is generated from the point and time of the potential collision.

Monte Carlo simulations have shown this simple method to work quite well, successfully avoiding collisions in each of several thousand simulations. However, it is desirable to develop a more rigorous approach based upon the same set-membership theory used in collision monitoring. In this context, a maneuver which guarantees no intersection between ellipsoids over a specified horizon could be planned.

The role of the *Detection Filter* agent is to detect failures in the Hall-effect thruster. The approach incorporates a nonlinear model of the system and compares the system's performance to that of the model. The model receives the same control inputs as the system. If a failure occurs, the outputs diverge. A specific combination of residuals identifies a failure in the thruster. Thus, when a failure occurs, the residual vector is fixed in direction, or at worst is fixed in a specific plane in the residual space. This greatly simplifies failure detection and generally eliminates the need for complex post-processing of the filter outputs for failure identification. The response to a detected thruster failure is part of the overall fault management plan, which is discussed in Section 5.

The role of the *Hardware Monitor* agent is to determine the maneuvering capability of the spacecraft. In order for a complete formation flying

maneuver to be carried out, the various sub-systems of the spacecraft must be operating within nominal bounds. For example, there must be sufficient power available to fire the HET, the HET must be operational and sufficiently cool, and the ADCS must be operating nominally. These represent a sample of the FFM dependencies on sub-systems. The full range and scope of sub-system dependencies has not been completely identified. The spacecraft is considered to be "delta-v operational" only if all of the dependent conditions are met. When this operational status changes, the *Hardware Monitor* agent notifies the *Constraint Manager*.

Formation Flying Agents

The formation flying agents implement the main functionality of the formation flying control system. This includes a series of coordinate transformations, maneuver planning, thruster alignment calculations, checking spacecraft constraints, and preparing hardware commands.

The *Coordinate Transform* agent first receives the absolute and relative position and velocity of both the reference spacecraft and itself, expressed in the ECI frame. It then performs a series of coordinate transformations to express these measured states in frames that can be used by other agents in the FFM.

The *Formation Planner* agent plans reconfiguration maneuvers in direct response to ground commands. The ground supplies the geometric goals of each relative satellite, defining the desired relative motion. The measured relative state is supplied by the *Coordinate Transform* agent. This information is used by the cluster supervisor to plan a multi-burn reconfiguration maneuver for all relative satellites.

The *Formation Keeper* agent is similar to the *Formation Planner* in that it implements the same control law, and may only plan maneuvers when it has the role of supervisor. The difference is that this agent autonomously plans formation keeping maneuvers when the specified deadband is exceeded. The deadband represents the allowable relative position error, and may be updated at any time from the ground.

The *Constraint Manager* agent receives a proposed delta-v sequence for one or more spacecraft, and prepares the necessary ADCS and HET commands to achieve these delta-v's. The proposed plan is ignored if a maneuver is already in progress, or if the *Hardware Monitor* has indicated that one or more of the spacecraft is not "delta-v operational."

An important role of the *Constraint Manager* is to ensure that the cluster is capable of achieving the

proposed maneuver prior to initiating the command sequence. Initiating a maneuver that cannot be completed due to spacecraft limitations (i.e., thermal or power restrictions) can result in relative motion that is extremely undesirable and potentially mission-threatening.

4. CONTROL SYSTEM DESIGN

The general control approach is outlined in Figure 4. The functionality represented in the upper box is carried out in the *Coordinate Transform* agent. The functionality shown in the lower box is implemented in both the *Formation Planner* and *Formation Keeper* agent.

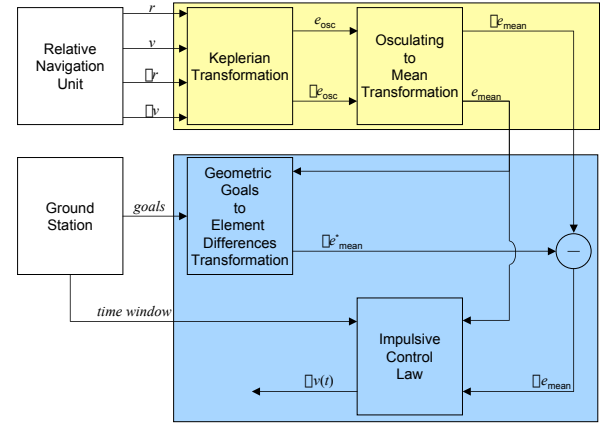


Figure 4: General Control Approach

Coordinate Frames and Transformations

The relative navigation unit provides an estimate of the absolute position (\mathbf{r}) and velocity (\mathbf{v}) of the reference spacecraft, as well as the relative position ($\Delta\mathbf{r}$) and velocity ($\Delta\mathbf{v}$) of the other spacecraft with respect to the reference. All measurements are provided in the inertial frame. A simple Keplerian transformation provides the osculating orbital elements of the reference (\mathbf{e}_{osc}) and the osculating orbital element differences ($\Delta\mathbf{e}_{osc}$). The elements are osculating at this point due to the J_2 perturbation. An osculating-to-mean transformation⁷ provides the mean reference elements (\mathbf{e}_{mean}) and the mean element differences ($\Delta\mathbf{e}_{mean}$).

The orbital element vector is defined as follows:

$$\mathbf{e} = [a \quad \varpi \quad i \quad q_1 \quad q_2 \quad \Omega]^T \quad (2)$$

where a is the semi-major axis, ϖ is the true argument of latitude, i is the inclination, and Ω is the right ascension. The dimensionless q -variables are defined as follows:

$$q_1 = e \cos \varpi \quad (3)$$

$$q_2 = e \sin \varpi \quad (4)$$

where e is the eccentricity and ϖ is the argument of perigee.

The ground station provides the geometric goals of the cluster (\mathbf{g}), as discussed in Section 2. This static goal set is transformed to desired element differences ($\Delta \mathbf{p}^*$) through the following procedure:

$$\begin{aligned} \Delta \mathbf{x}_H^* &= f_{gx}(\mathbf{e}_{mean}, \mathbf{g}) \\ \Delta \mathbf{p}^* &= f_{xe}(\mathbf{e}_{mean}, \Delta \mathbf{x}_H^*) \end{aligned} \quad (5)$$

where f_{gx} is the transformation function from geometric goals to Hill's frame coordinates, and f_{xe} is the transformation function from Hill's frame coordinates to element differences⁷. The f_{gx} transformation is expressed as:

$$\begin{aligned} x_H &= \frac{1}{2} a_E \cos(\varpi_0 + \varpi) \\ y_H &= a_E \sin(\varpi_0 + \varpi) + y_0 \\ z_H &= z_i \sin \varpi \cos \varpi \\ \dot{x}_H &= \frac{1}{2} a_E n \sin(\varpi_0 + \varpi) \\ \dot{y}_H &= a_E n \cos(\varpi_0 + \varpi) \\ \dot{z}_H &= n \varpi (z_i \cos \varpi + z_0 \sin \varpi) \end{aligned} \quad (6)$$

where the angle ϖ_0 is termed the complementary phase angle. It is related to ϖ_0 through the equation

$$\varpi_0 = \sin^{-1} \left[\frac{\sin \varpi_0}{\sqrt{4 \varpi^2 \sin^2 \varpi_0}} \right] \quad (7)$$

The physical meaning of ϖ_0 is best described through the diagram in Figure 5.

The ellipse represents the actual motion of the spacecraft in the relative frame. The superimposed circle represents the evolution of the absolute orbit. As the orbit is traversed, the relative spacecraft follows the path of the 2x1 ellipse. The angles are related by the fact that the y-component of the circle and ellipse are always equal.

Impulsive Control Law

The control law is an impulsive approach initially designed by Alfried⁸ and further developed by Mueller. The approach is derived from Gauss' variational equations, which provide the expressions

for the instantaneous change in orbital element differences due to an impulse. A closed-form solution is obtained which expresses a finite delta-v sequence as a function of the errors in orbital element differences. The derivation assumes zero eccentricity and does not account for the J_2 perturbation.

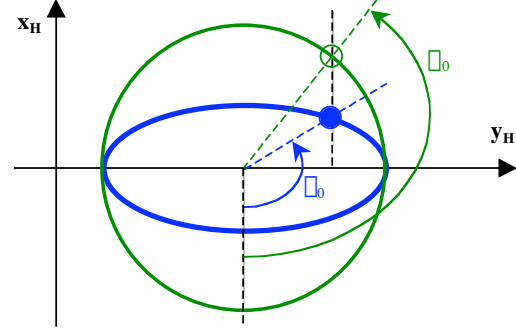


Figure 5: Physical relationship between $\Delta \mathbf{p}_0$ and $\Delta \mathbf{p}_0$

The element error ($\Delta \mathbf{e}_{mean}$) is computed as the difference between the desired and measured element differences.

In general, the burn sequence consists of three in-plane delta-v's and a single out-of-plane delta-v. The out-of-plane delta-v is given by the equation

$$\Delta v_n = \frac{h}{r} \sqrt{(\Delta i)^2 + (\Delta \varpi \sin i)^2} \quad (8)$$

where h is the angular momentum and r is the magnitude of the position vector. This delta-v is applied at the following argument of latitude:

$$\varpi = \tan^{-1} \left[\frac{\Delta \varpi \sin i}{\Delta i} \right] \quad (9)$$

The immediate goal of the out-of-plane burn is to adjust the cross-track amplitude by changing i and ϖ . It is clear from Gauss's variational equations, however, that this out-of-plane burn will also affect the elements q_1 , q_2 and ϖ . These cross-coupling effects are added to the initial errors before computing the in-plane burns, as follows:

$$\Delta q_1 = \Delta q_1 + \frac{r}{p} q_2 \sin \varpi \cot i \Delta v_n \quad (10)$$

$$\Delta q_2 = \Delta q_2 + \frac{r}{p} q_1 \sin \varpi \cot i \Delta v_n \quad (11)$$

$$\Delta p = \Delta p + \frac{r}{h} \sin \Delta \cot i \Delta v_n \quad (12)$$

where p is the semi-latus rectum.

The required set of in-plane delta-v's is given by the following equations.

$$\Delta v_1 = \frac{na}{3N\Delta} \left[\Delta \Delta_0 + \frac{3}{2} \Delta \bar{a} (\Delta_1 - \Delta_0) + 2 \Delta \tilde{q}_0 \right] + \frac{na}{4} \left[\frac{M}{N} \Delta q + \frac{M}{N} + 1 \right] \Delta \bar{a} \quad (13)$$

$$\Delta v_2 = \frac{na}{4} (\Delta q - \Delta \bar{a}) \quad (14)$$

$$\Delta v_3 = \Delta \frac{na}{3N\Delta} \left[\Delta \Delta_0 + \frac{3}{2} \Delta \bar{a} (\Delta_1 - \Delta_0) + 2 \Delta \tilde{q}_0 \right] + \frac{na}{4} \left[\frac{M}{N} \Delta q + \frac{M}{N} + 1 \right] \Delta \bar{a} \quad (15)$$

where n is the mean orbit rate, a is the semi-major axis, Δ_0 is the current argument of latitude, and Δ_1 is the argument of latitude at which the first burn is to be applied.

$$\Delta_1 = \tan^{-1} \left[\frac{\Delta q_2}{\Delta q_1} \right] \quad (16)$$

Furthermore, we have

$$\Delta q = \Delta q_1 \cos \Delta_1 + \Delta q_2 \sin \Delta_1 \quad (17)$$

$$\Delta \tilde{q}_0 = \Delta q_1 \sin \Delta_0 + \Delta q_2 \cos \Delta_0 \quad (18)$$

$$\Delta \bar{a} = \frac{\Delta a}{a} \quad (19)$$

The parameters M and N are maneuver time variables. Each is a positive integer, with M representing the number of half-orbits between the 1st and 2nd burn, and N the number of half-orbits between the 1st and 3rd burn. They are subject to the following constraints:

$$\begin{aligned} N &\geq M + 1 \\ N &\text{ even} \\ M &\text{ odd} \end{aligned} \quad (20)$$

The *time window* supplied by the ground station (see Figure 4) indicates the minimum and maximum number of orbits that the maneuver may last. The M

and N parameters are varied within this specified time window to find the combination which requires the smallest delta-v.

It is interesting to note a few observations about the sensitivity of the delta-v to the maneuver time variables. Only the first and third delta-v burns are dependent upon M and N . The magnitude of the third delta-v is inversely proportional to N . Thus, as the maneuver duration increases, the size of the third burn approaches zero. This can be a useful fact for planning maneuvers in which limited power is available for the final burn.

The total delta-v savings can be seen in the sum of the absolute values of the first and third burns. For leader-follower resizing maneuvers, the expression reduces to:

$$|\Delta v_1| + |\Delta v_3| = \frac{2na}{3N\Delta} |\Delta \Delta| \quad (21)$$

In this maneuver, the second delta-v is zero. Therefore, the total required delta-v is inversely proportional to N . When the nature of the motion is considered, this result makes perfect sense. An along-track distance of Δy is covered in time Δt , with two equal and opposite burns applied at the beginning and end of the maneuver. The first burn creates a difference in the semi-major axis, Δa , which results in along-track drift. The final burn sets Δa back to zero, eliminating the drift. The change in semi-major axis corresponds to a difference in the mean orbit rate, n , expressed as:

$$\Delta n = \frac{3}{2} \frac{n}{a} \Delta a \quad (22)$$

This represents the relative drift rate. The final along-track offset is found to be:

$$\Delta y = \Delta \frac{3}{2} n \Delta a (\Delta t) \quad (23)$$

This equation shows that the same amount of along-track offset can be achieved with a smaller Δa by allowing more time to complete the maneuver.

Another interesting result is found when we consider a reconfiguration from IPLF to CIPE, for the special case where $y_0 = a_E$. In this case, the size of the first and third delta-v's reduces to:

$$|\Delta v_1| + |\Delta v_3| = \frac{na}{8} |\Delta \Delta| \quad (24)$$

Thus, the required delta-v is independent of the maneuver time variables. For the more general case of $y_0 \neq a_E$, however, significant fuel savings can be achieved with the proper selection of M and N . Consider a reconfiguration from a 300 m offset IPLF formation to a CIPE formation with $a_E = 60$ m. Two separate cases of the reconfiguration are shown below.

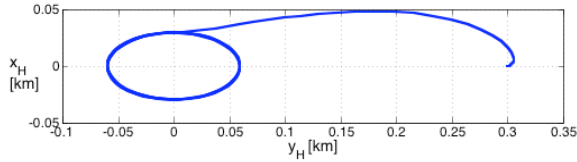


Figure 6: IPLF to CIPE Reconfiguration, 1 orbit

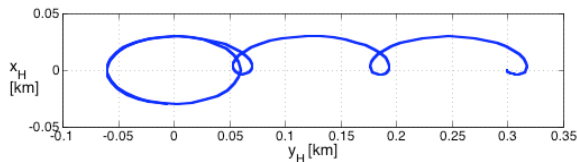


Figure 7: IPLF to CIPE Reconfiguration, 3 orbits

The single orbit maneuver requires 0.048 m/s of delta-v, whereas the 3-orbit maneuver requires only 0.016 m/s. The dependence of the total delta-v on the M and N parameters is shown in Figure 8. This plot nicely illustrates the fuel savings that can be achieved by choosing the burn times appropriately. Recall that M indicates the number of half-orbits between the 1st and 2nd burn. If the second burn is applied one or three half-orbits after the first, then the optimal delta-v can never be achieved. By allowing more time between the 1st and 2nd burn, the spacecraft is able to drift at a slower rate, which requires less delta-v.

Spacecraft Constraints

The physical constraints of the spacecraft can have a significant impact on the control system design. Consider the thruster related constraints, for example. Since each spacecraft is outfitted with a single thruster that is fixed to the bus, it is necessary to slew the entire spacecraft prior to each burn such that the thruster is pointed in the proper direction. Thermal constraints on the HET limit the frequency and duration of thruster firings, and operating the HET requires significant power.

Considering these constraints, in order to schedule a burn one must first ensure that sufficient power will be available, that the HET will be sufficiently cool, and that the ADCS will have had enough time to slew to the delta-v orientation. If these conditions cannot be guaranteed, then there is no guarantee that the proposed burn will be successfully applied. As

previously discussed, canceling a burn in the midst of a maneuver can have serious consequences.

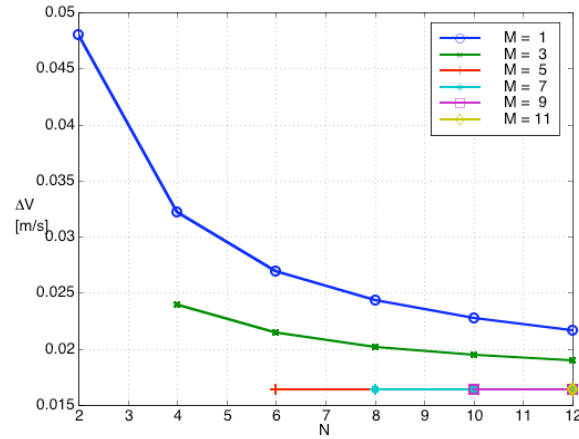


Figure 8: Delta-V Dependency on Burn Times

In general, the thermal constraints on the HET impose a duration limit of 10 minutes on any single burn. If two burns are applied close together, however, this limit must certainly be reduced. From a formation flying perspective, the constraint is most meaningful if expressed as a function of the previous burn duration and the time between burns. Thruster performance is typically not presented in this fashion, however, and therefore requires collaboration between the control system designers and the thruster manufacturer.

In order to satisfy the power constraints, the spacecraft must remain in a sun-pointing mode for a sufficient period of time. Since a maneuver requires the re-orientation of the spacecraft, it impacts the amount of available power. Thus, a constraint which must be adhered to in maneuver planning is directly coupled with the maneuver itself. The simplest way to resolve this issue is to impose conservative power-related constraints on the planning process and ignore the coupling. Otherwise, a detailed model of the power profile as a function of time-on-sun and other factors must be generated and used within the planning algorithm. A conservative approach was chosen for TechSat 21 given the challenges in obtaining such a model.

Another interesting constraint involves the pointing of the star-tracker while aligning the thruster for a burn. The star-tracker sensor is required for precise attitude determination. Pointing it at bright objects causes the sensor to turn off temporarily, thereby reducing the accuracy of the attitude determination. When the thruster is fired, it is important to have the least amount of attitude error possible to avoid subsequent errors in the relative motion. Therefore, it

is necessary for the star-tracker to continue functioning nominally prior to and during every burn.

This constraint required the development of a separate algorithm to determine an orientation that properly aligns the thruster while pointing the star-tracker as far away as possible from the brightest natural objects in the sky—the sun, earth and moon.

For each commanded burn, a specific orientation must be chosen so that the appropriate commands may be sent to the ADCS. The ADCS may be commanded to track a specified orientation within the local-vertical local-horizon (LVLH) frame. Prior to the start-time of the burn, the ADCS is commanded to transition to “LVLH Tracking” mode, and to stay in this mode throughout the duration of the burn. In order to command this mode, two parameters must be supplied: 1) the body vector to be aligned with the velocity, and 2) the body vector to be aligned with nadir. These two body vectors provide sufficient information to define a desired orientation that is fixed with respect to the local frame.

The solution is found subject to two constraints. First, the bore-sight of the thruster nozzle must be pointed in the opposite direction of the desired delta-v. Second, the bore-sight of the star-tracker camera must be pointed as far away as possible from the sun, earth and moon.[□]

An infinite number of orientations exist which satisfy the first constraint. This can be envisioned by first aligning the thruster in the desired direction, then rotating the spacecraft about the bore-sight of the thruster nozzle. A discretized search is performed in this fashion, where the average angular distance between the star-tracker bore-sight and the three bright bodies is calculated at each step. The orientation with the largest average angular distance is chosen.

5. FAULT MANAGEMENT DESIGN

Fault management design for a cluster of spacecraft is characterized by a higher level of complexity than that for a single spacecraft because it must account for the effect of faults on the entire cluster in addition to accounting for the effects on the individual spacecraft. The various types of spacecraft faults can be generally categorized into two basic groups: spacecraft-level faults and cluster-level faults. Spacecraft-level faults can be handled by the individual spacecraft without

affecting other members of the cluster. Cluster-level faults impact the performance of the cluster as a whole, and may require additional action by at least one other member of the cluster.

The classification of a fault as spacecraft-level or cluster-level is highly dependent on the design of the particular spacecraft. For example, a thruster failure may mean a cluster level fault on a spacecraft with a single thruster, but it may be only a spacecraft-level fault if the spacecraft has multiple thrusters and is capable of compensating for the loss of one. Generally, though, failures in the attitude control, propulsion, inter-satellite communication and relative navigation subsystems are most likely to produce cluster-level faults.

The purpose of the cluster fault management system is to first identify and then respond to cluster-level faults. The primary objectives are to ensure the safety of the cluster and to maintain its overall functionality. In traditional designs, a typical response to most faults is to enter safemode. However, this is not necessarily the safest response to cluster-level faults. It is therefore important to define a cluster-level safemode.

The goal of safemode is to prevent the spacecraft from incurring damage. It is generally designed to use a minimum amount of power and collect a maximum amount of energy. When a spacecraft is a member of a cluster, however, the impact on cluster performance and safety must also be considered. As such, the ISL and the relative navigation subsystems should be considered critical subsystems and should be active during safemode. Enabling communication between a safed spacecraft and the rest of the cluster allows the cluster to remain aware of the safed spacecraft's position, thus making collision monitoring and avoidance more effective and allowing the cluster to maintain formation for a longer period of time. This translates to an easier recovery of full cluster functionality should the safed spacecraft return to operational mode.

In the case of TechSat 21 spacecraft, safemode does not keep the ISL operational. Therefore, any spacecraft entering safemode must be considered lost. In order to ease recovery of full functionality, the cluster uses the last known location of the spacecraft in the collision monitoring algorithms and maintains the current formation as long as possible. However, the relative position uncertainty of the safed spacecraft increases while the ISL remains down, and eventually the cluster is forced to make collision avoidance maneuvers.

[□] It should be noted that reflections off of other spacecraft may be sufficiently bright so as to interfere with the star-tracker. These reflections are not yet taken into account by the algorithm.

Collision monitoring and collision avoidance are essential to the safety of a cluster. Under nominal operations, when the control systems of cluster members are fully operational, the threat of collision is minimal. In the event of a cluster-level failure, however, the performance of relative estimation and control are degraded. It is imperative during these times to have a robust method for detecting possible collisions and a feasible approach for avoiding them. The collision avoidance approach must balance the objectives of safety and performance. Factors such as maintaining cluster functionality, permanently removing a failed spacecraft as a collision threat, equitable distribution of fuel consumption among cluster members and spacecraft system constraints should form part of the collision avoidance system of an operational cluster. The methods of collision monitoring and collision avoidance developed for the TechSat 21 cluster are briefly described in Section 3.

One of the expected benefits of the cluster paradigm is its robustness to single-point failures. Maintaining cluster functionality in the face of full or partial failures of cluster members is therefore a primary objective of the cluster fault management plan. These adjustments may be designed to be made autonomously or to require ground intervention. In the case of TechSat 21 the high level decisions to maintain cluster functionality are made outside of the FFM, but the features that enable the execution of those decisions are incorporated into the FFM. For example, if one of the members were to experience a HET failure, the FFM would detect the anomalies through a detection filter. Because of the experimental status of the FFM, the only action it would take would be to report the anomaly to the ground. Once the ground verified the failure, it would notify the cluster and the FFM would then adjust by making the spacecraft with the failed thruster become the reference, since the reference does not need to maneuver.

6. CURRENT STATUS & FUTURE WORK

A preliminary build of the Formation Flying Module has been completed and delivered to AFRL with a variety of closed-loop control demonstrations included. Four different formation flying scenarios have been developed in which the following features are demonstrated:

- In-Plane Reconfiguration
- Out-of-Plane Reconfiguration
- Formation Keeping
- Collision Monitoring and Avoidance
- Reference Rollover
- Supervisor Rollover

In every scenario, the FFM agent code runs as three separate executables on a Linux workstation while the simulation of the three-satellite cluster is conducted using MultiVehicleSim™ on an Apple PowerBook. Each ObjectAgent application connects to the simulation via TCP/IP. Commands for the FFM are entered at the command line and sent as ObjectAgent messages to the Interface Agent of each spacecraft.

One scenario involves both a reference rollover and a supervisor rollover, followed by an out-of-plane reconfiguration. The three satellites – named Athos, Porthos and Aramis – are initialized in a 100 meter Leader-Follower formation. Porthos lies in the center, and initially serves as both the reference and the supervisor. The reference orbit and spacecraft model used in the scenario are consistent with the information outlined at the beginning of Section 2.

Several steps are taken in the command sequence. First, the reconfiguration goals are supplied. This command supplies the necessary information to define the desired relative motion of each spacecraft. In this case, Porthos is desired to have a 250 meter along-track offset and a 25 meter cross-track amplitude, while Athos is desired to have a 500 meter along-track offset and a 50 meter cross-track amplitude. The reconfiguration goals are not supplied for Aramis, because it will soon become the new reference.

Table 2: Reconfiguration Goals

	Athos	Porthos	Aramis
y_0	500 m	250 m	-
a_E	0 m	0 m	-
\square_b	-	-	-
z_i	0	0	-
z_{\square}	50 m	25 m	-

The next command in the sequence is a reference rollover. This command is sent to all spacecraft, identifying Aramis as the new cluster reference. The *Coordinate Transform* agent receives the command, and begins to treat Aramis as the origin of the relative frame when computing the mean orbital element differences.

Following the reference rollover, a command is issued for a supervisor rollover. Here, the identity of the supervisor changes from Porthos to Athos. The *Formation Planner* and *Formation Keeper* agents on all spacecraft are notified. From this point forward, only Athos may plan formation keeping or reconfiguration maneuvers.

Finally, the reconfiguration is commanded. The estimated mean orbital elements are provided from

the *Coordinate Transform* agent with Aramis treated as the reference. The *Formation Planner* agent on Athos transforms the supplied reconfiguration goals into desired element differences. The error between the measured and desired element differences is supplied to the control law to compute the required delta-v sequence. This proposed plan involves a multiple burn sequence for both Porthos and Athos. The plan is sent to the *Constraint Manager* agent, which computes the required orientation of each spacecraft for each burn, along with the set of times at which the HET must be turned on and off.

The target attitude for each burn is achieved by means of a separate attitude control system which was developed specifically for FFM validation. Perfect state knowledge and actuation is used, as the objective is to validate the overall software functionality rather than the algorithm performance. Figure 9 illustrates the successful achievement of the desired formation geometry.

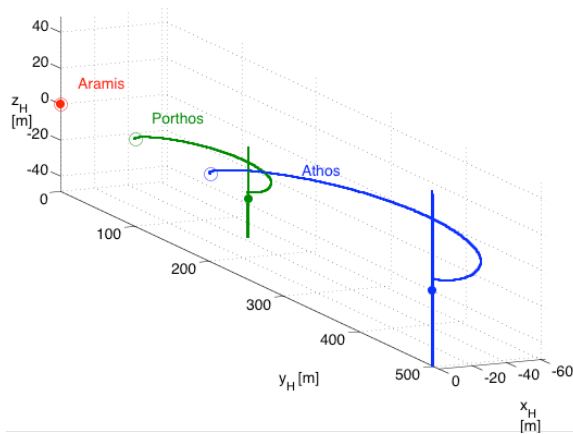


Figure 9: IPLF to OOPLF Reconfiguration

The Air Force Research Laboratory has discontinued the TechSat 21 program. However, the experience of designing this control system within the context of an evolving mission profile and subject to the constraints of an existing spacecraft design has brought increased awareness of the complex issues inherent to formation flying. The sensitivity of mission performance to relative navigation accuracy, the requirement of a cluster-level safemode, and the need for detailed knowledge of subsystem dynamics for accurate maneuver planning are a few of the most important issues discovered.

7. ACKNOWLEDGMENTS

This work has been supported by two United States Air Force SBIR Phase II contracts from the Surveillance and Control Division of the Air Force Research Laboratory's Space Vehicles Directorate. The contract numbers are F29601-02-C-0029 and

F29601-01-C-0134 and the program manager is Paul Zetocho.

8. REFERENCES

- [1] Schetter, T. P., M. E. Campbell, and D. M. Surka, "Comparison of Multiple Agent-based Organizations for Satellite Constellations," *2000 FLAIRS AI Conference Proceedings*, Orlando, Florida, May 2000.
- [2] Schetter, T. P., M. E. Campbell, and D. M. Surka, "Multiple Agent-Based Autonomy for Satellite Constellations," *Second International Symposium on Agent Systems and Applications Proceedings*, Zurich, Switzerland, September 2000.
- [3] Surka, D. M., M. Brito, and C. G. Harvey, "The Real-Time ObjectAgent Software Architecture for Distributed Satellite Systems", *2001 IEEE Aerospace Conference Proceedings*, Big Sky, Montana, March 2001.
- [4] Mueller, J. B., D. Surka, B. Udrea, "Agent-Based Control of Multiple Satellite Formation Flying", *The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: A New Space Odyssey*, Montreal, Canada, June 2001.
- [5] Thomas, S. J., J. B. Mueller, D. M. Surka, C. G. Harvey, "Monitoring and Analysis of Multiple Agent Systems", *1st Workshop on Radical Agent Concepts*, Goddard Space Flight Center, January 2001.
- [6] Zetocho, P. and M. C. Brito, "Development of a Testbed for Distributed Satellite Command and Control," *2001 IEEE Aerospace Conference Proceedings*, Big Sky, Montana, March 2001.
- [7] Gim, D. W. and K. T. Alfriend, "The State Transition Matrix of Relative Motion for the Perturbed Non-Circular Reference Orbit," AAS 01-222, *AAS/AIAA Space Flight Mechanics Meeting*, Santa Barbara, CA, February 2001.
- [8] McLaughlin, C., K. T. Alfriend, and T. A. Lovell, "Analysis of Reconfiguration Algorithms for Formation Flying Experiments", *International Symposium for Formation Flying Missions and Technologies*, October 2002, Toulouse, France.
- [9] Campbell, M. and B. Udrea, "Collision Avoidance in Satellite Clusters", *American Control Conference*, Anchorage AK, May 2002. Submitted to the *AIAA Journal of Guidance, Control, and Dynamics*.